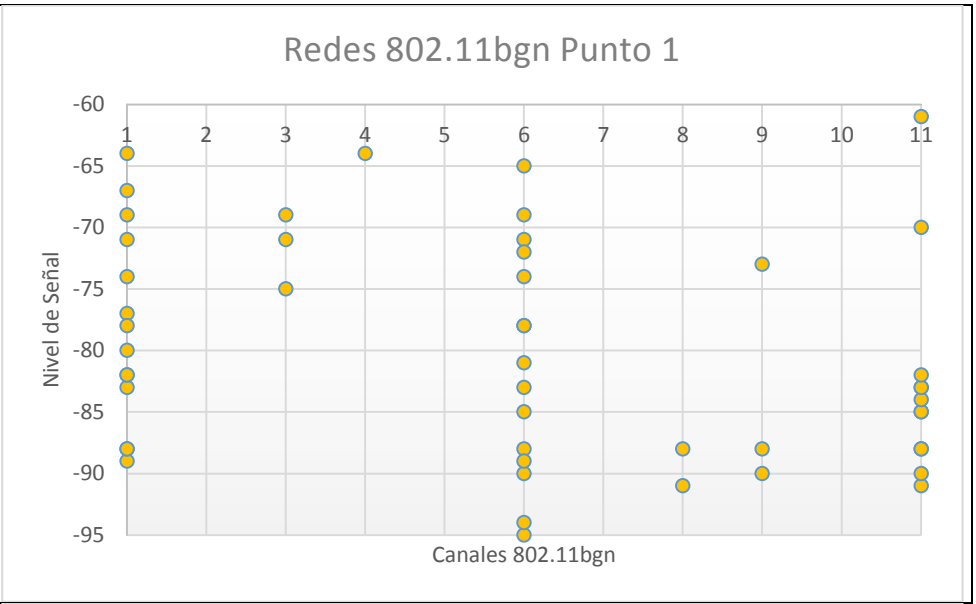
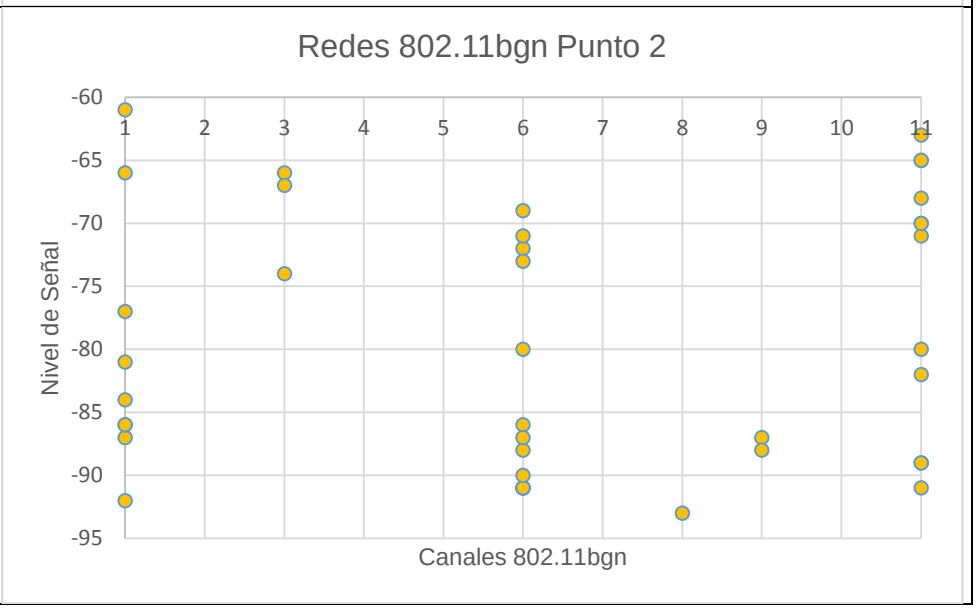
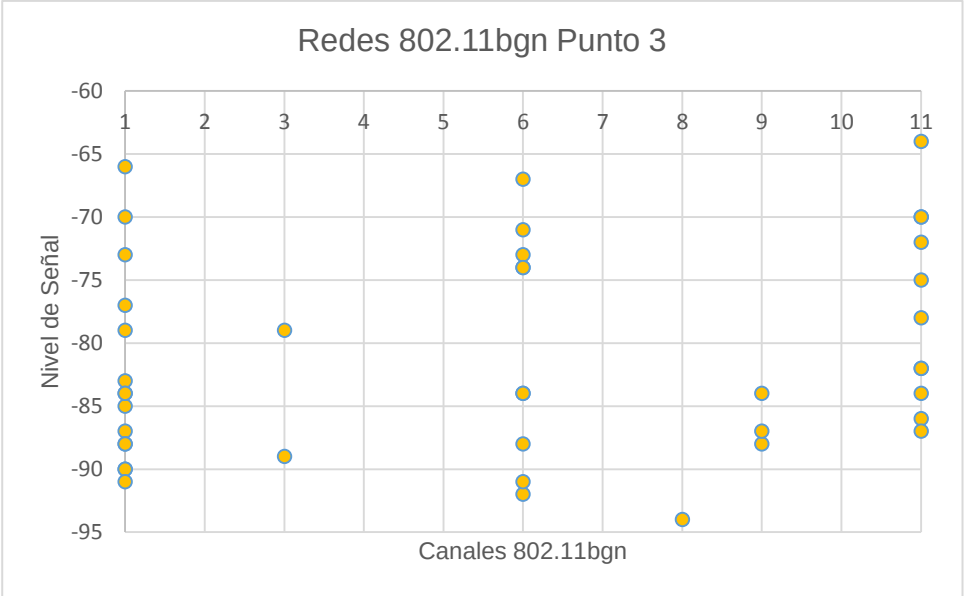
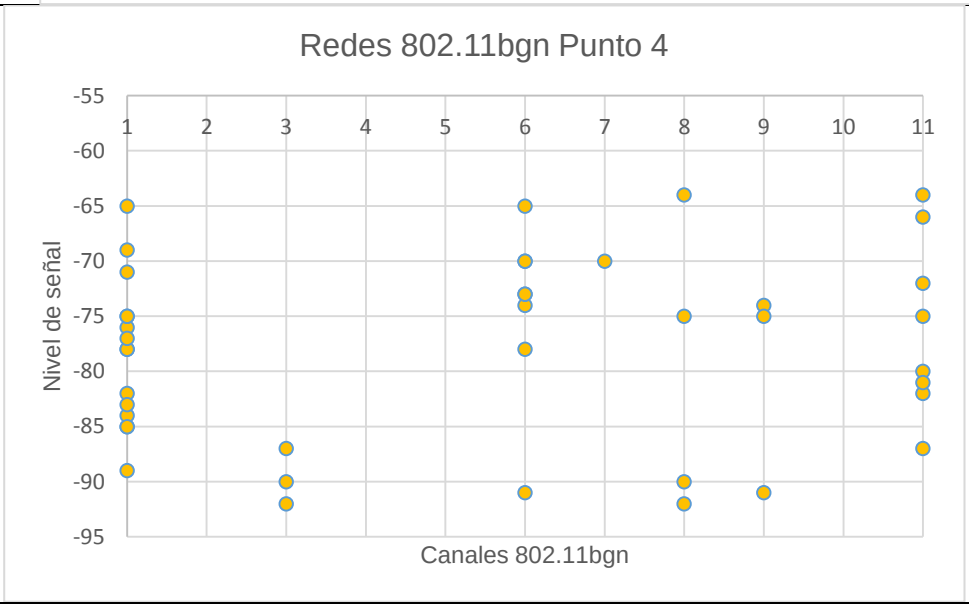
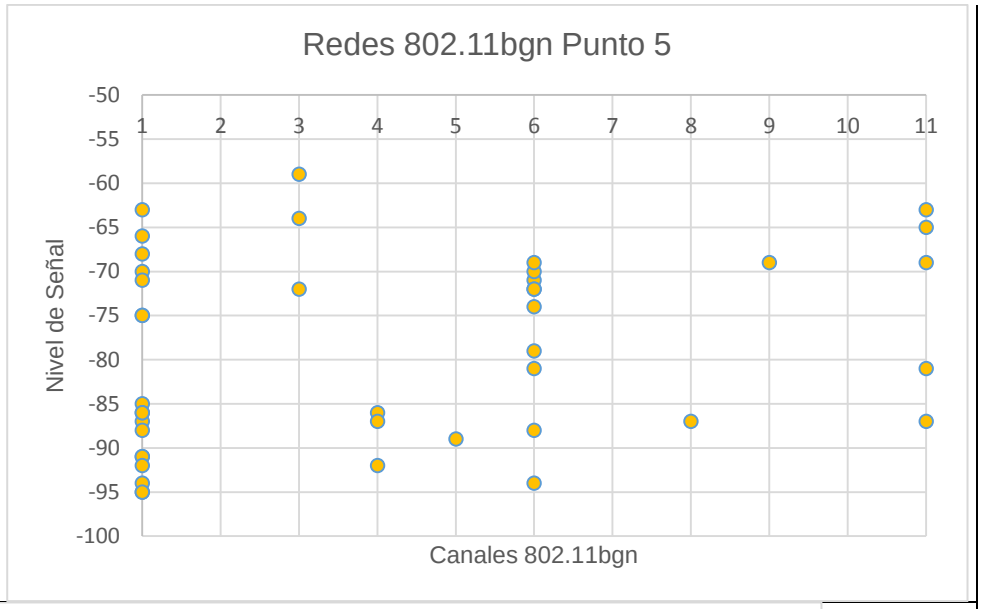
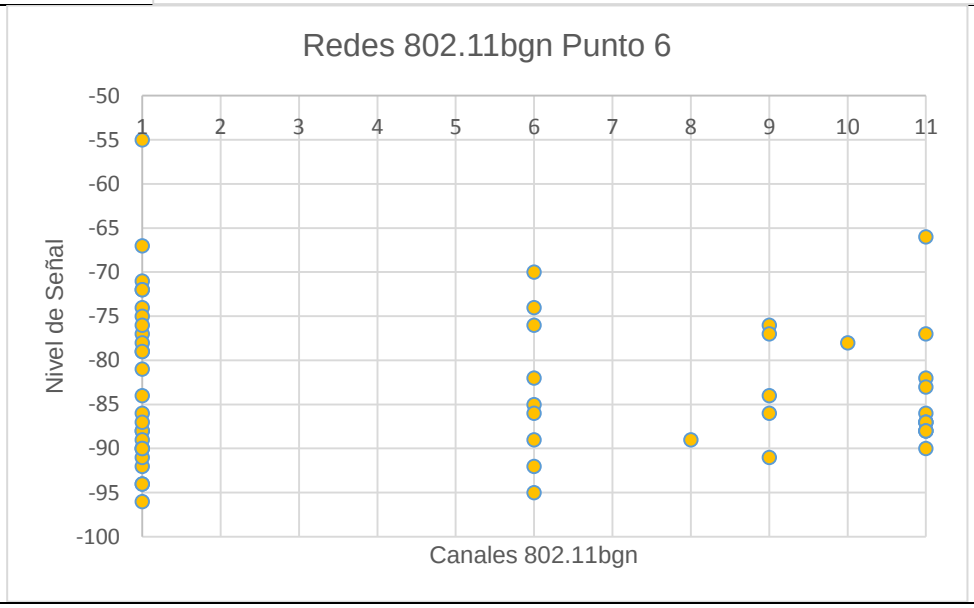
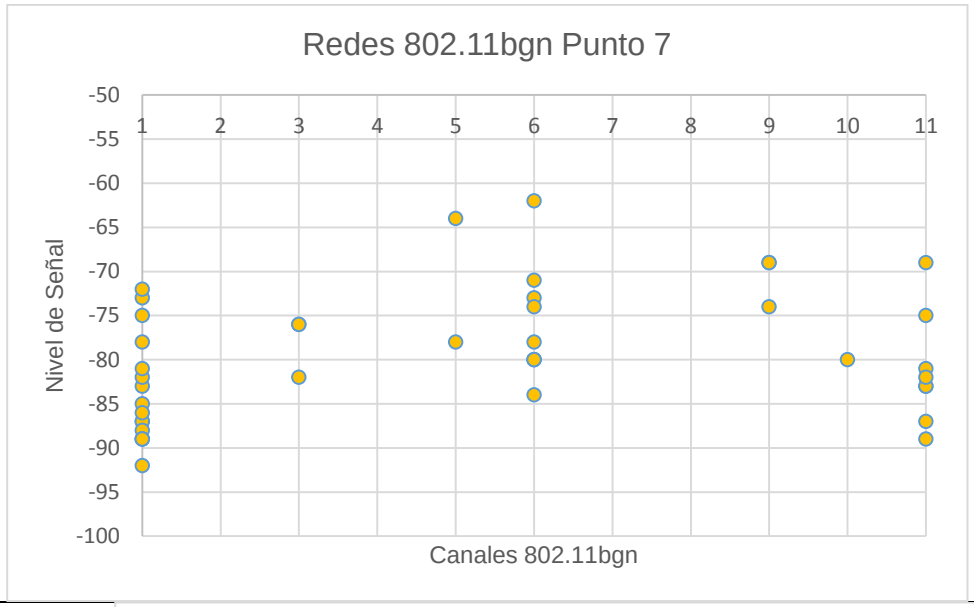
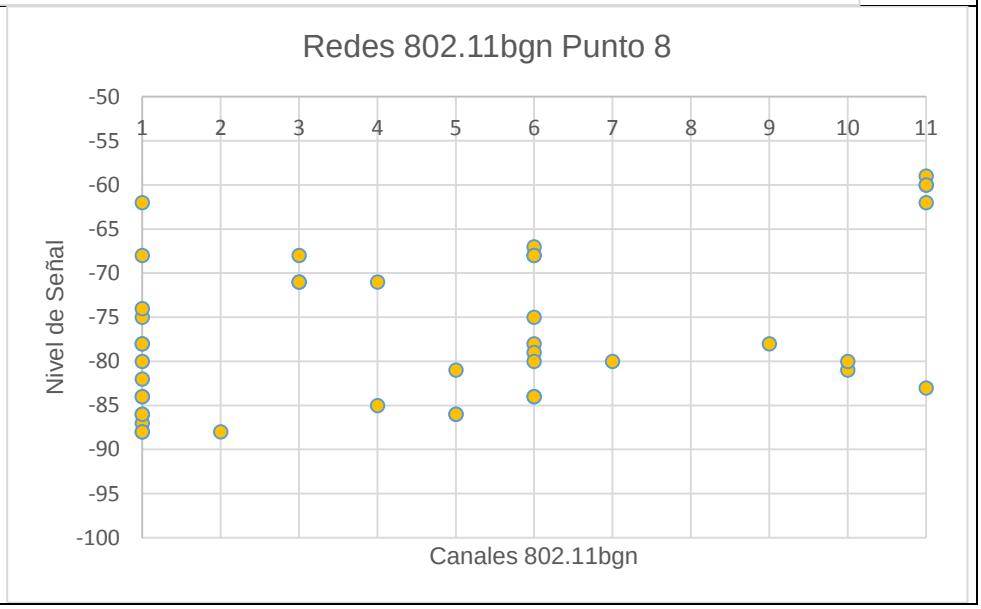
Anexos

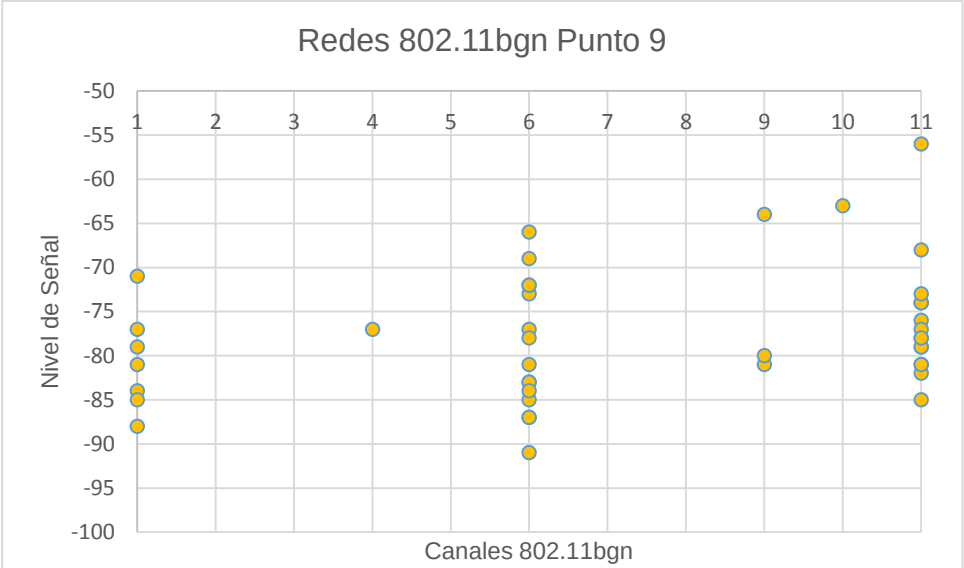
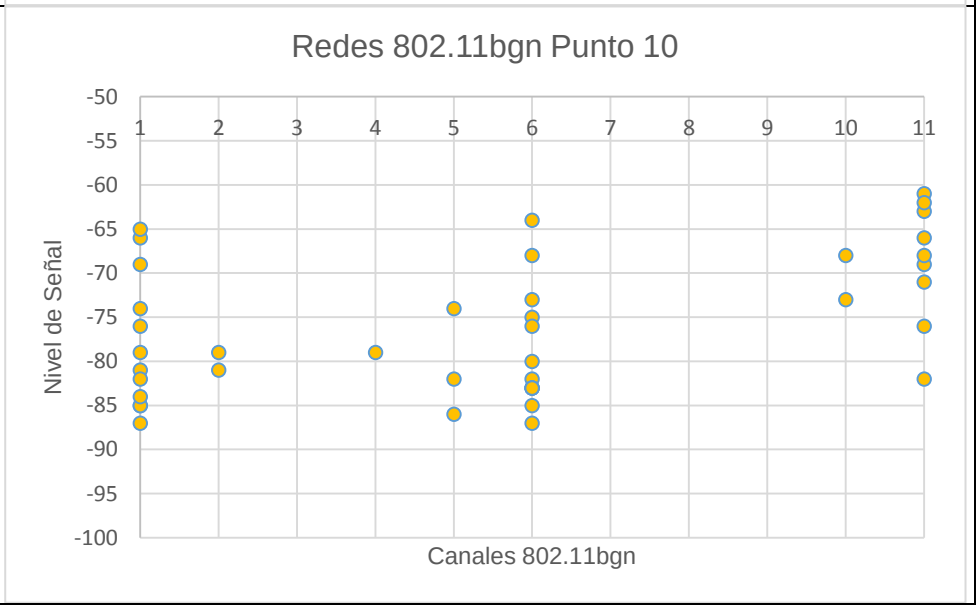
1. Intensidad de señal de los AP en la Av. La Mar por cada punto de medición.

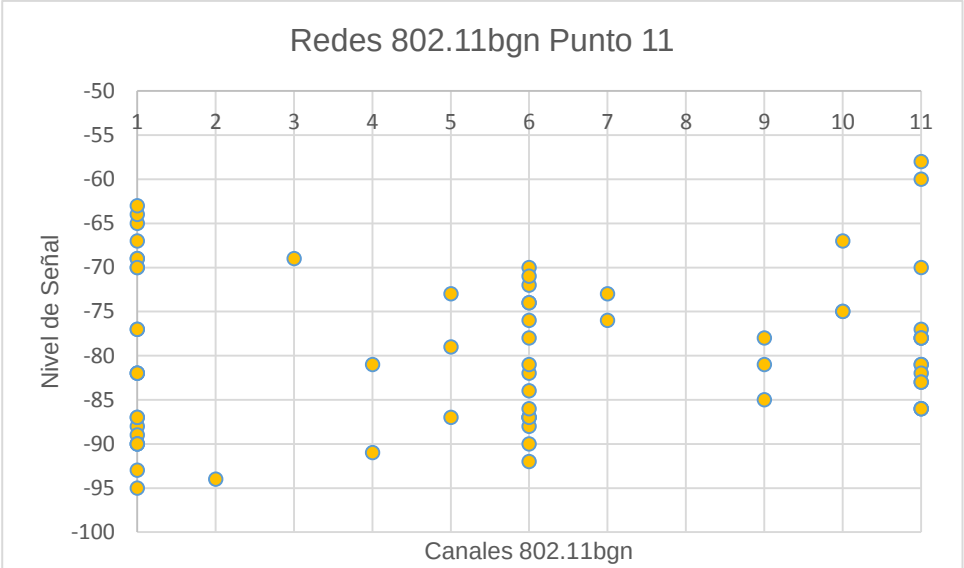
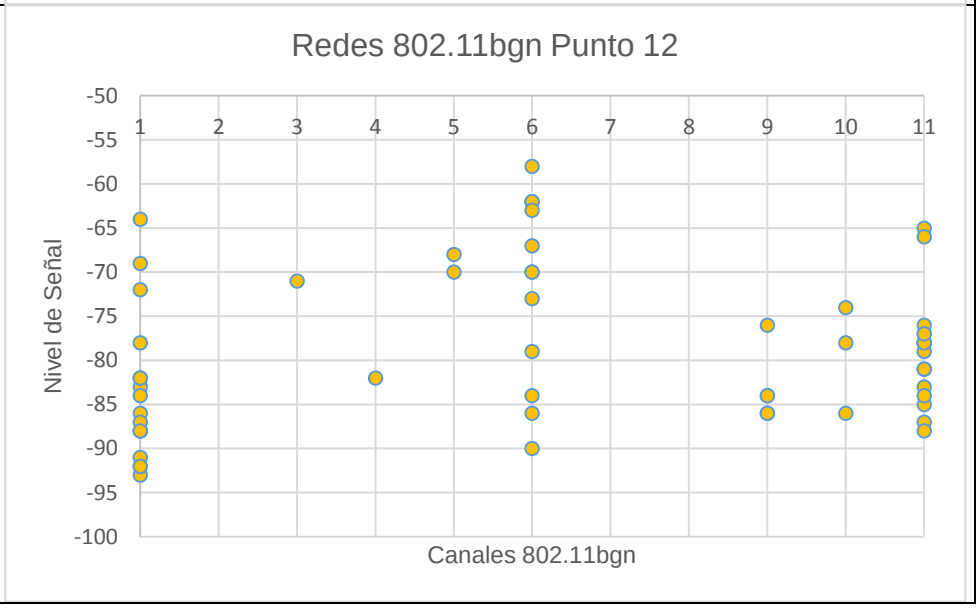
P1		
P2		

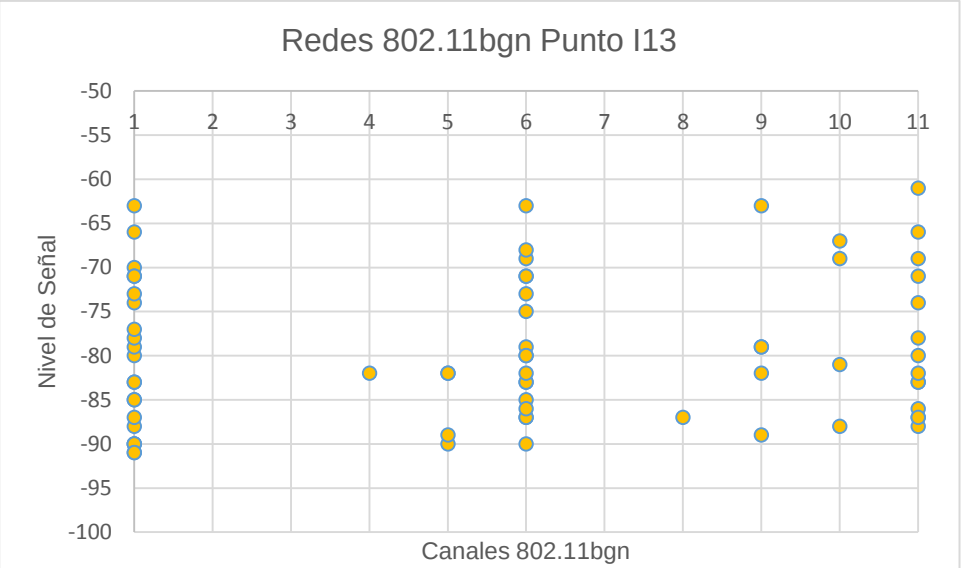
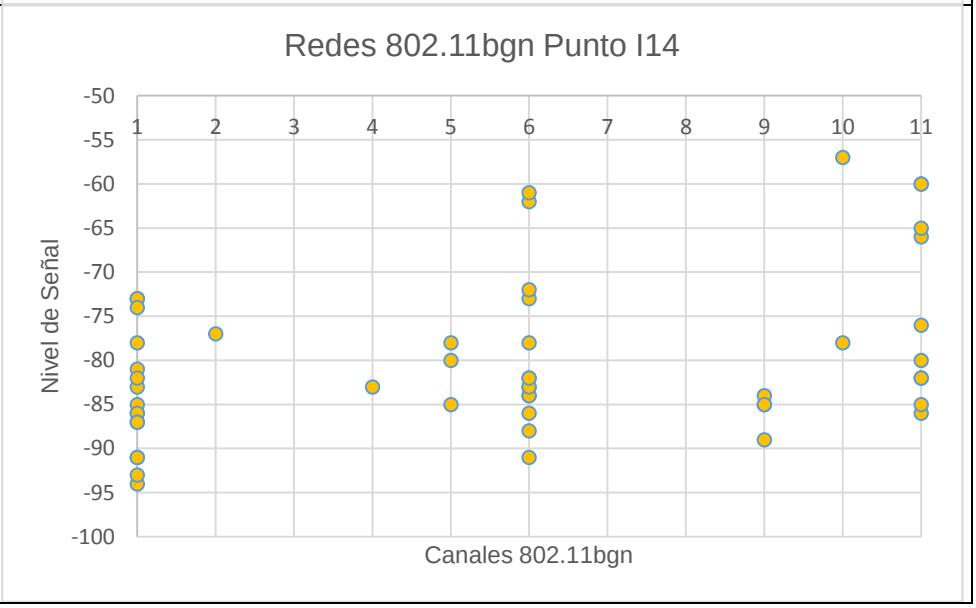
P3		 Redes 802.11bgn Punto 3 Nivel de Señal -60 -65 -70 -75 -80 -85 -90 -95 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P4		 Redes 802.11bgn Punto 4 Nivel de señal -55 -60 -65 -70 -75 -80 -85 -90 -95 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

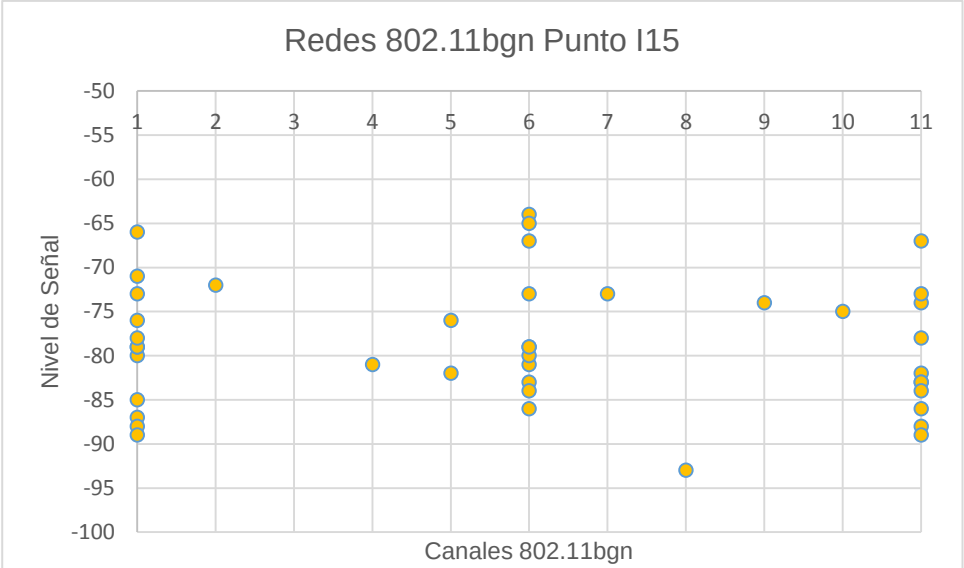
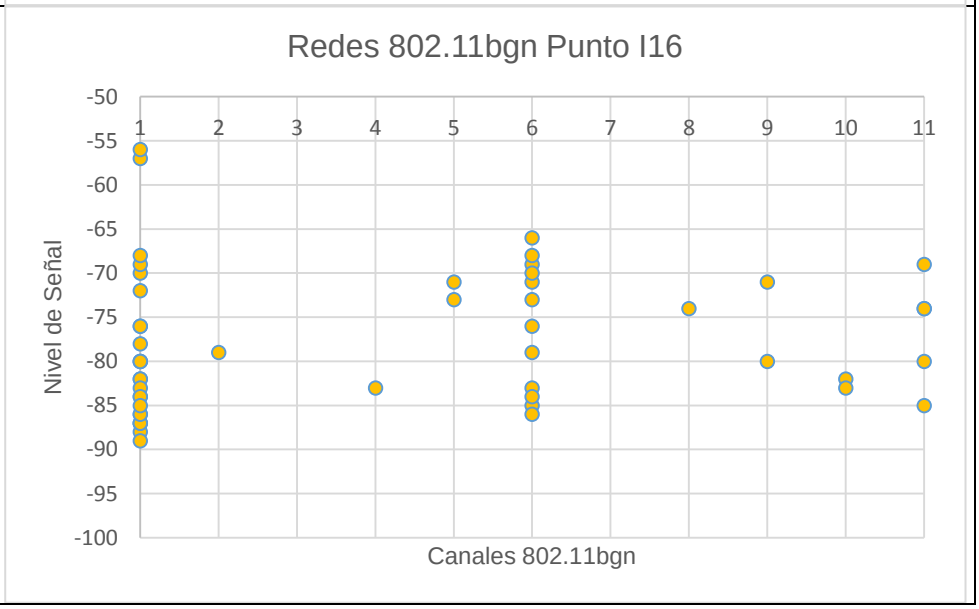
P5	 Redes 802.11bgn Punto 5 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P6	 Redes 802.11bgn Punto 6 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

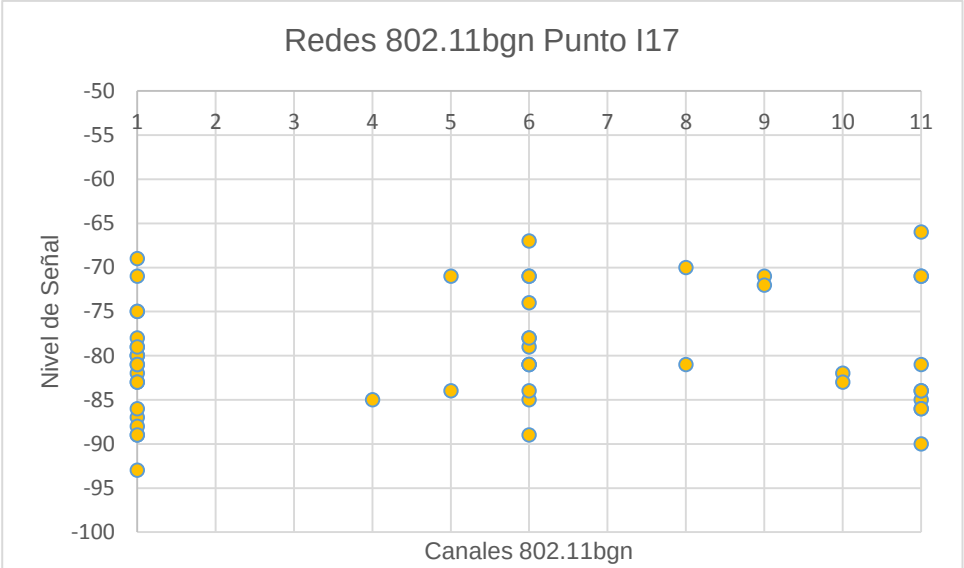
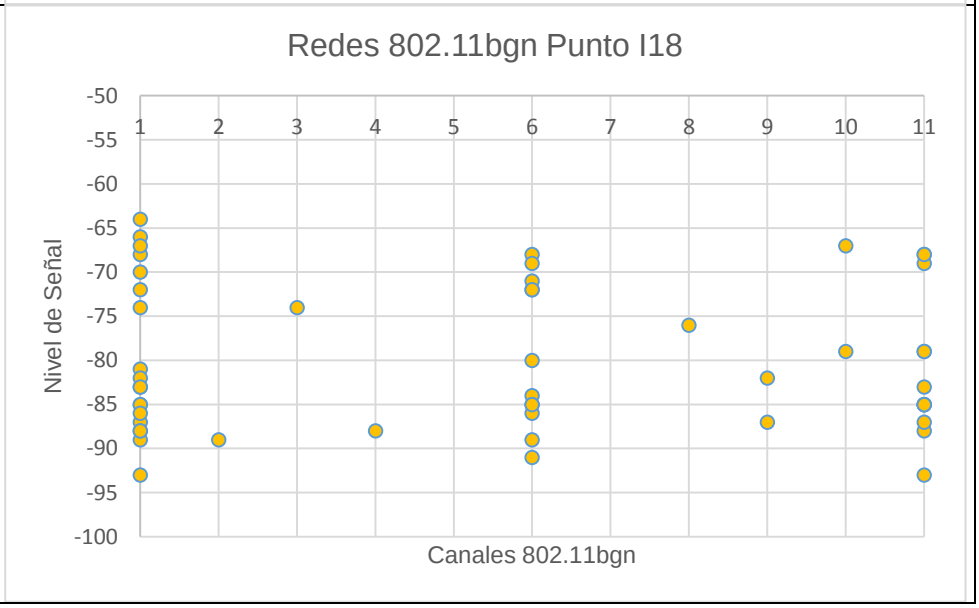
P7	 Redes 802.11bgn Punto 7 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P8	 Redes 802.11bgn Punto 8 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

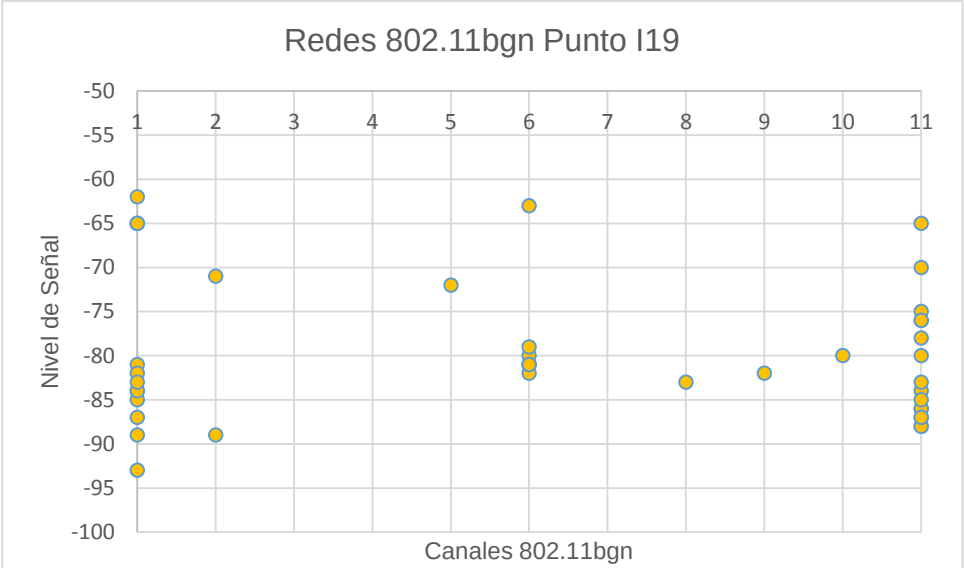
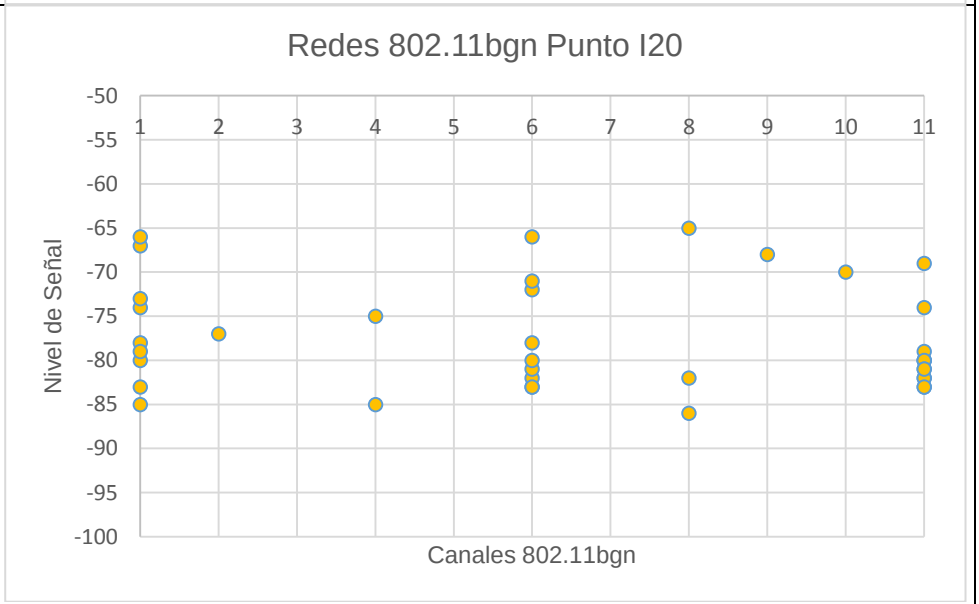
P9		 Redes 802.11bgn Punto 9 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P10		 Redes 802.11bgn Punto 10 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

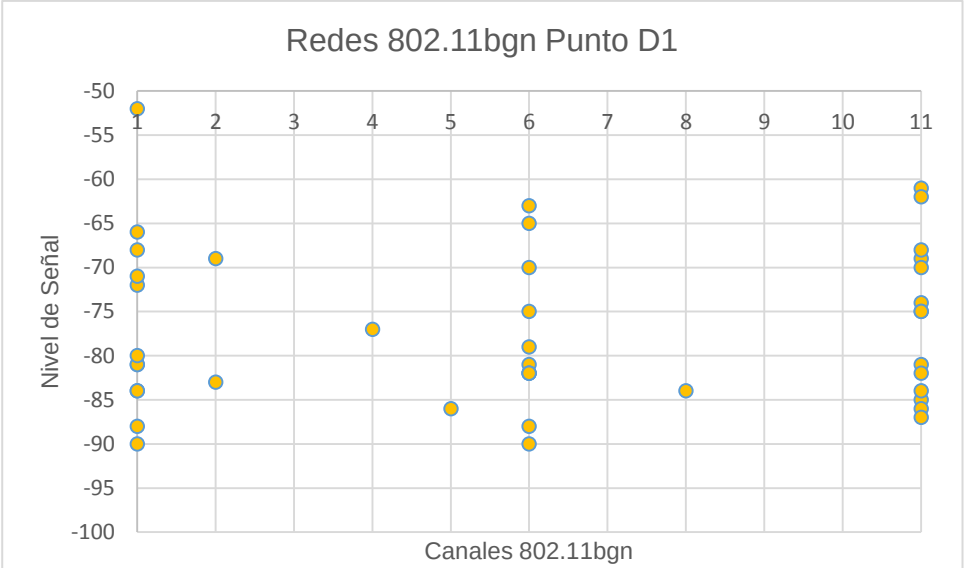
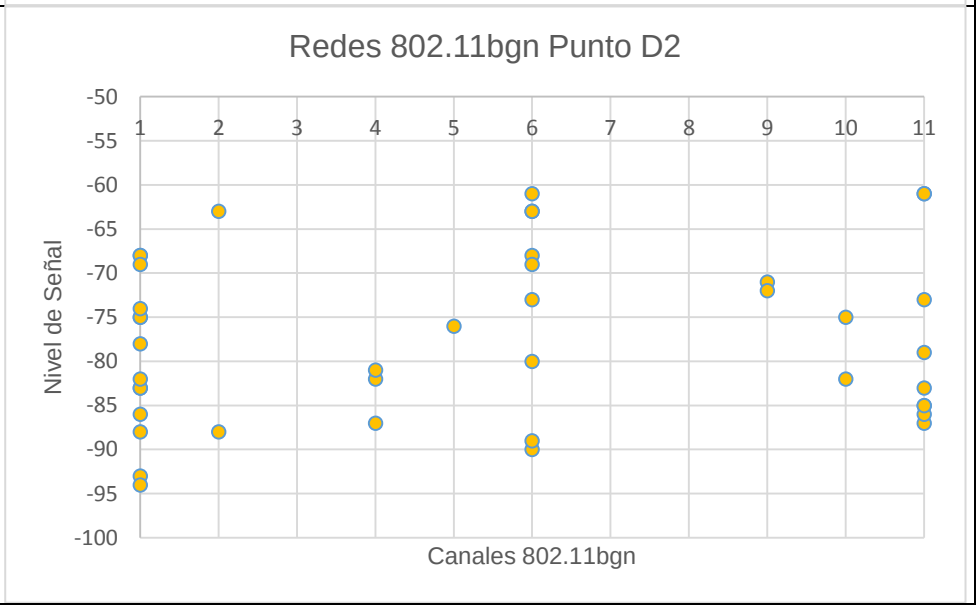
P11		 Redes 802.11bgn Punto 11 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn
P12		 Redes 802.11bgn Punto 12 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn

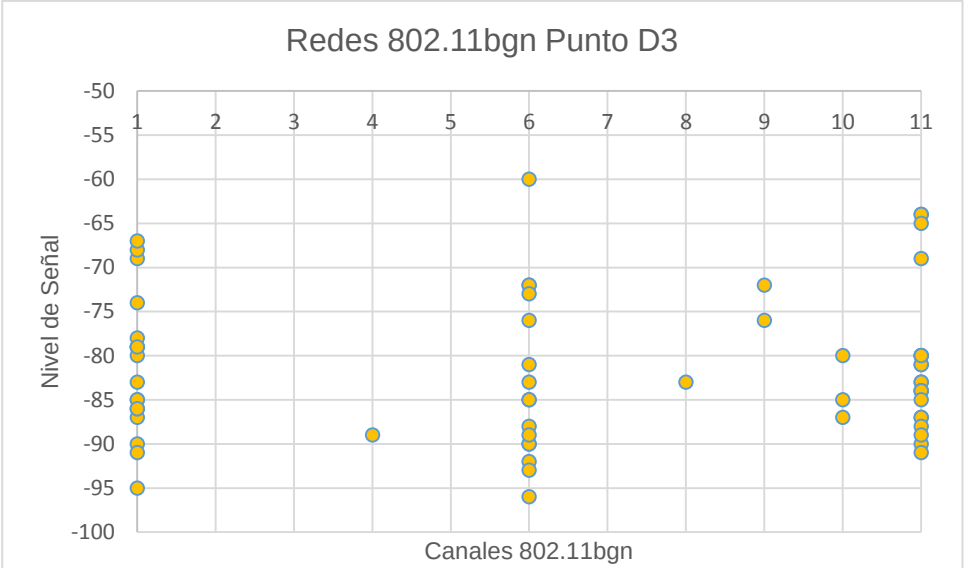
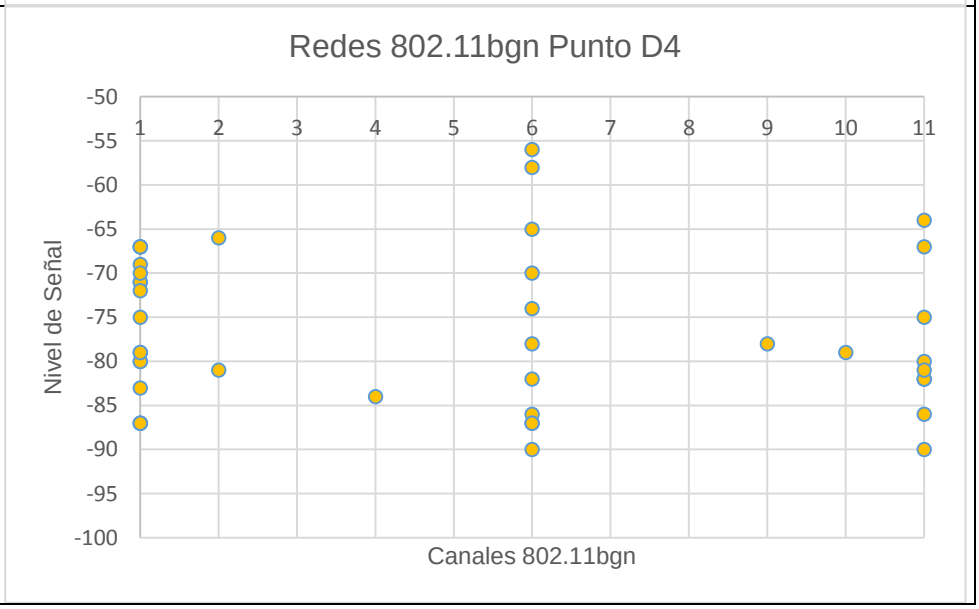
P13		 Redes 802.11bgn Punto I13 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P14		 Redes 802.11bgn Punto I14 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

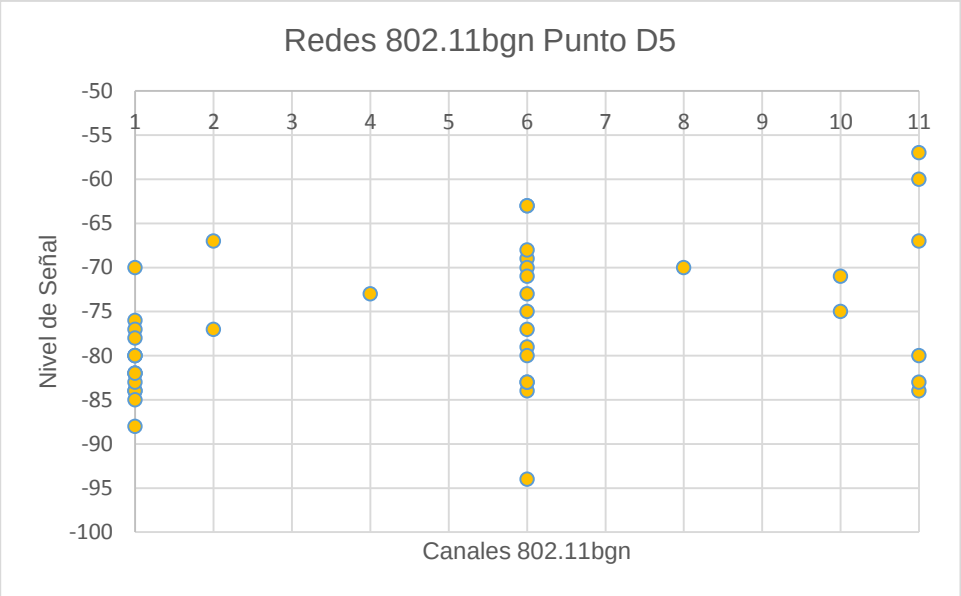
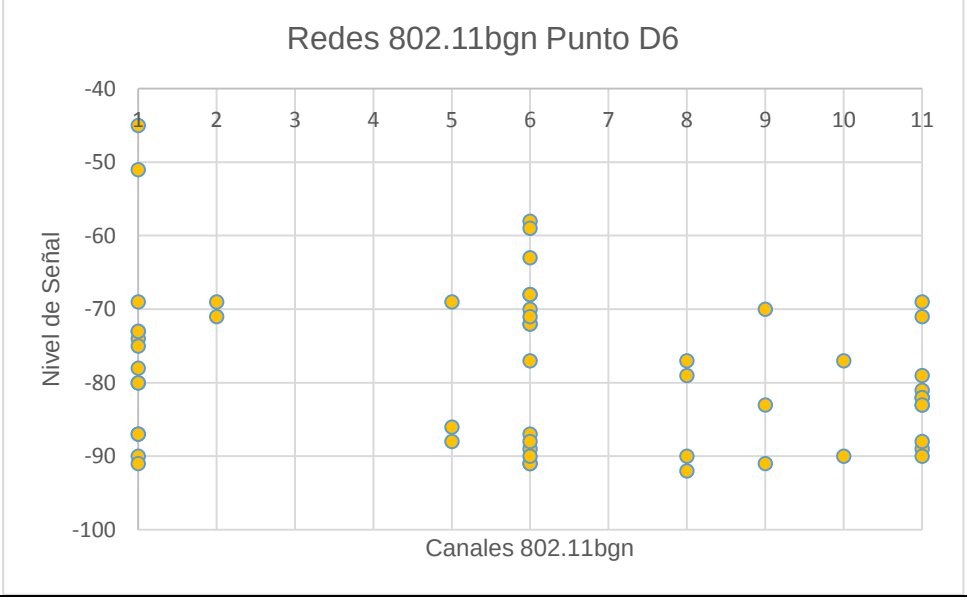
P15		 Redes 802.11bgn Punto I15 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 Nivel de Señal 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P16		 Redes 802.11bgn Punto I16 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 Nivel de Señal 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

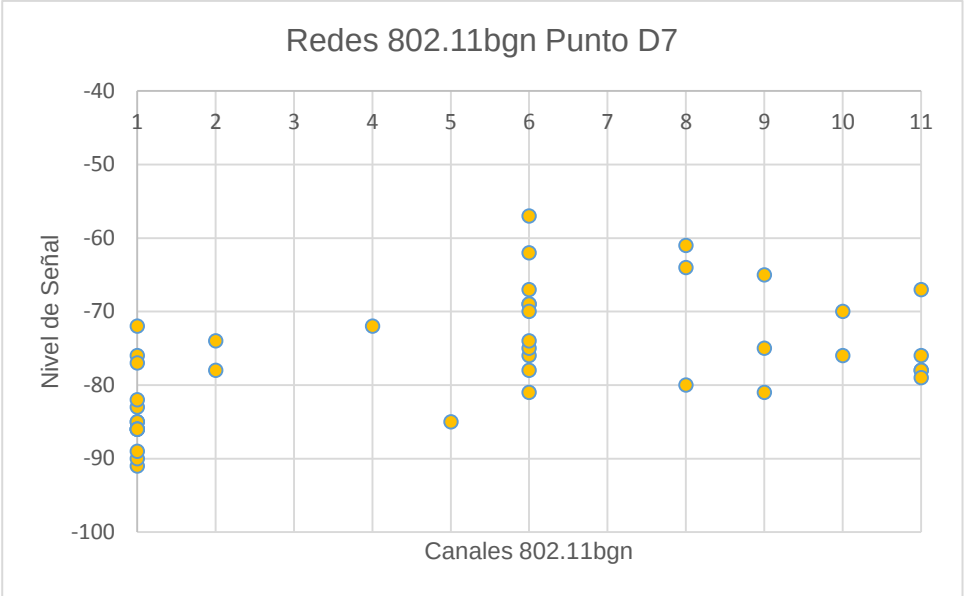
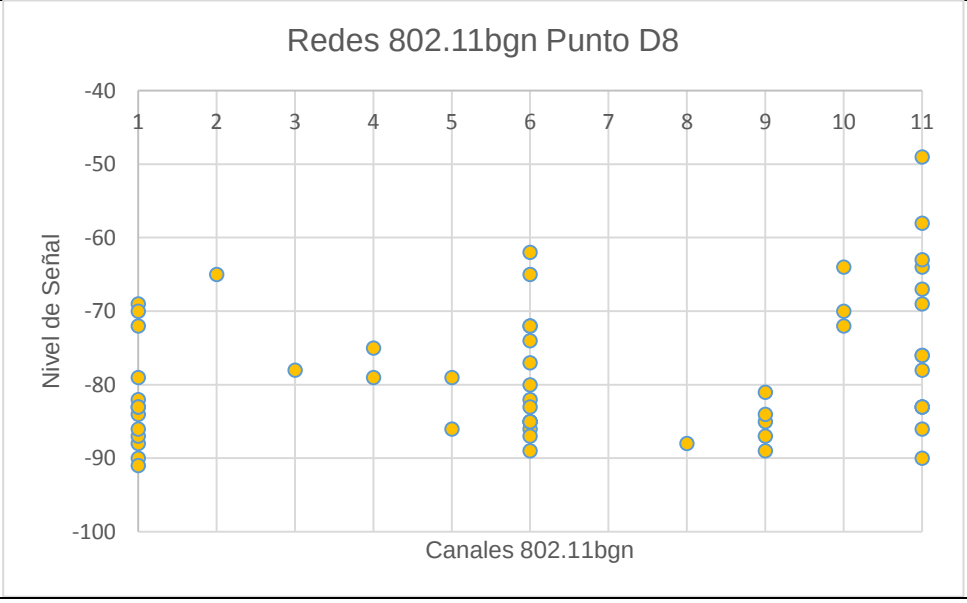
P17		 Redes 802.11bgn Punto I17 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn
P18		 Redes 802.11bgn Punto I18 -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn

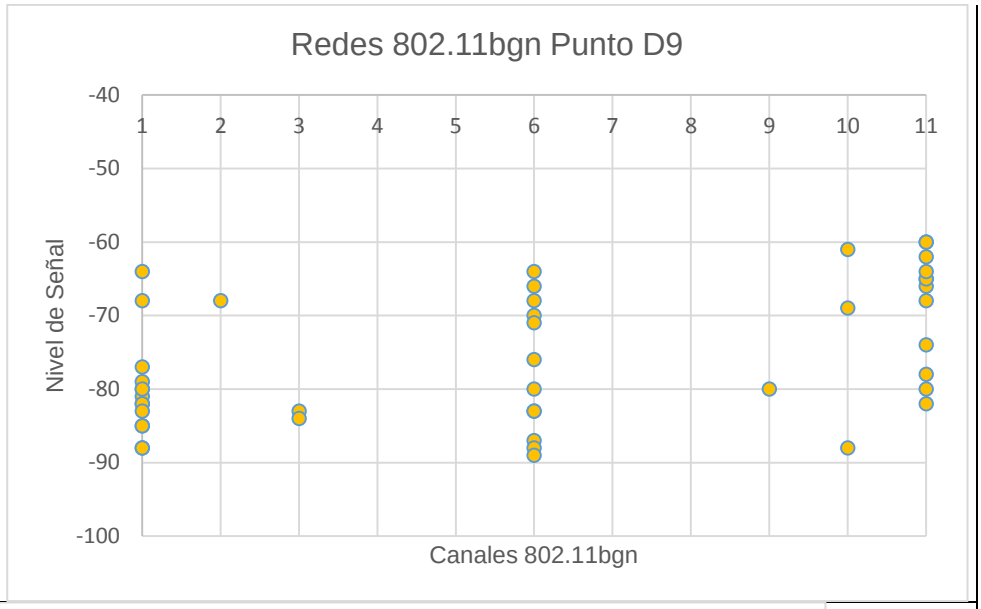
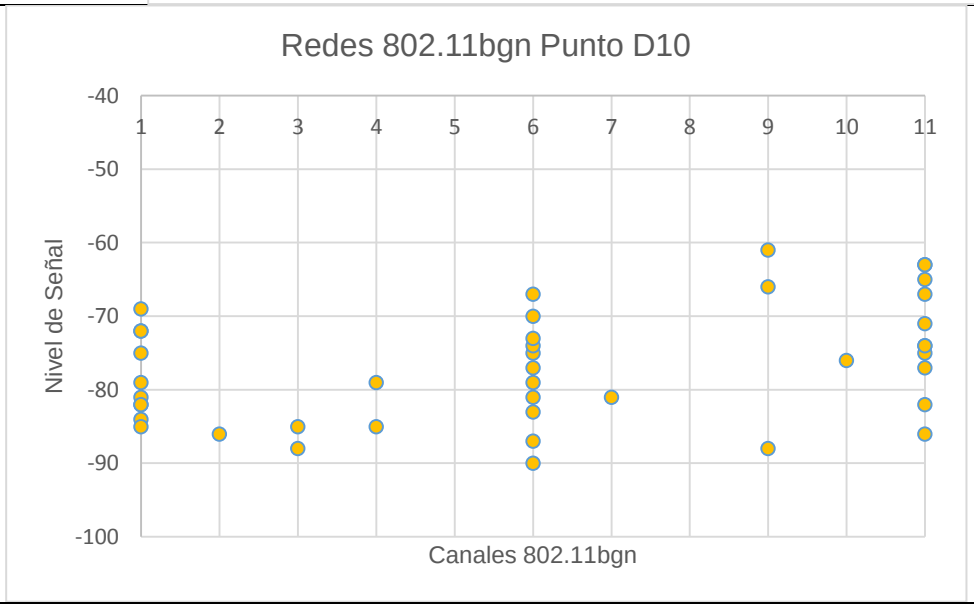
P19		 Redes 802.11bgn Punto I19 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P20		 Redes 802.11bgn Punto I20 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

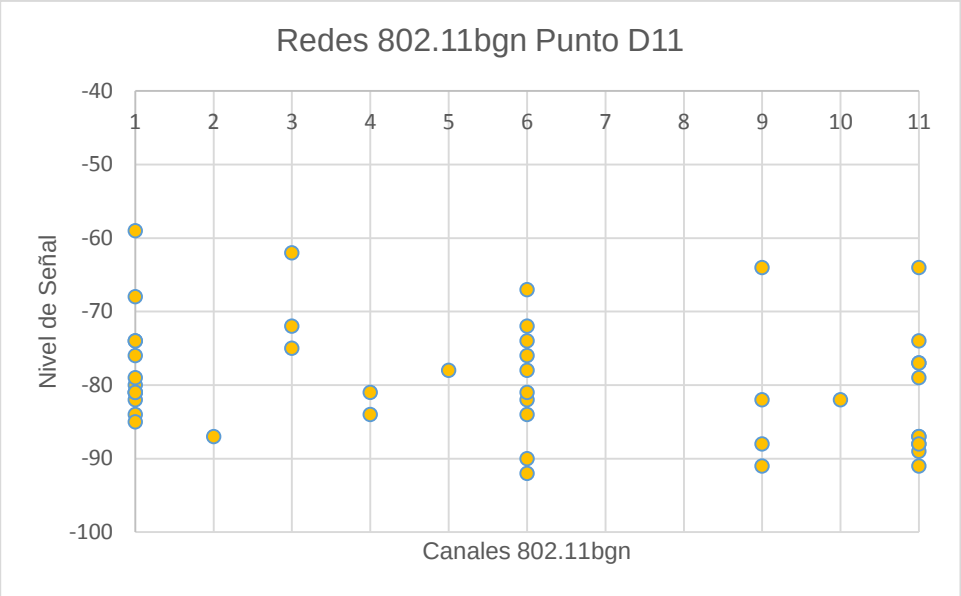
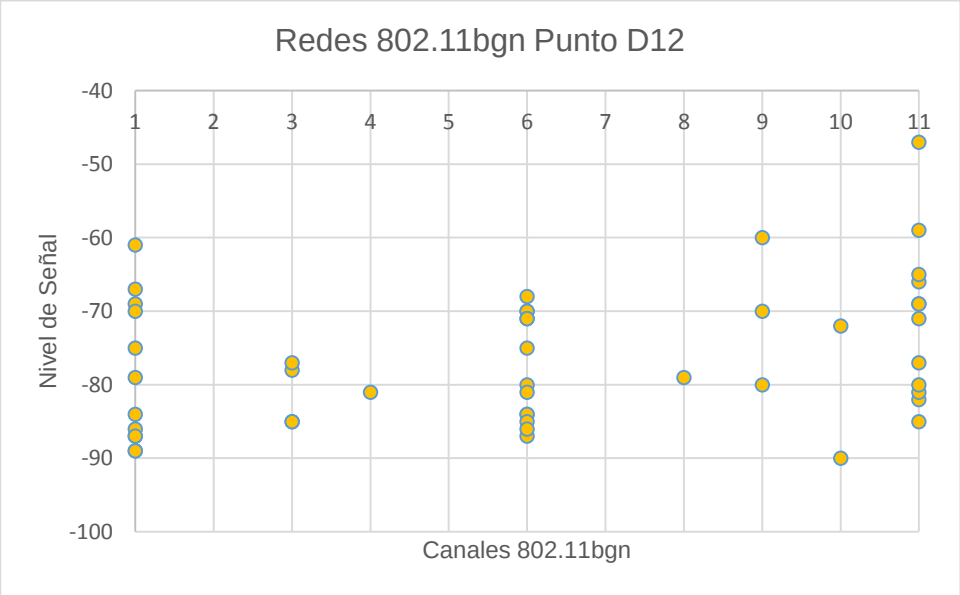
P21		 Redes 802.11bgn Punto D1 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P22		 Redes 802.11bgn Punto D2 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

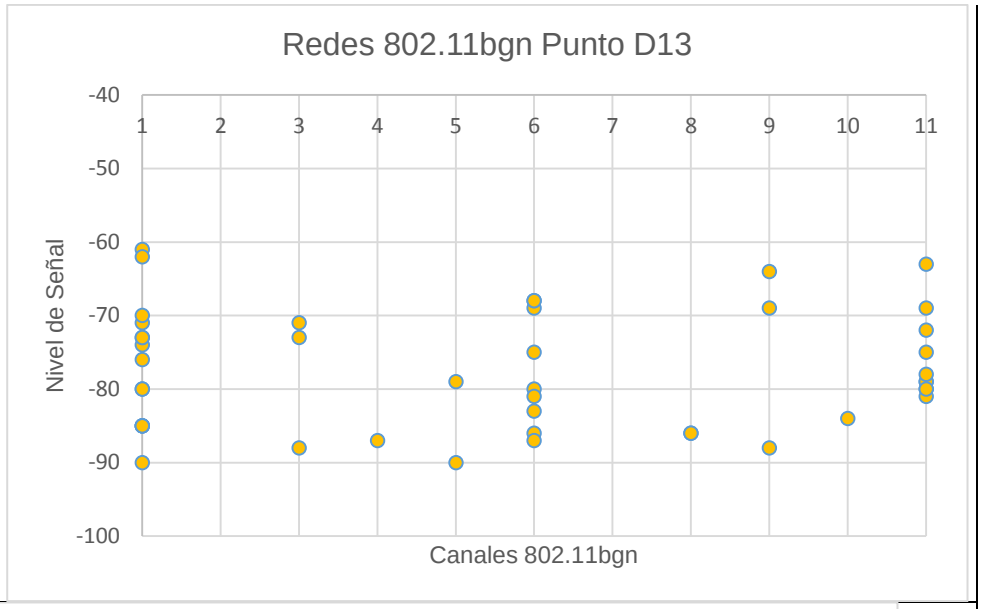
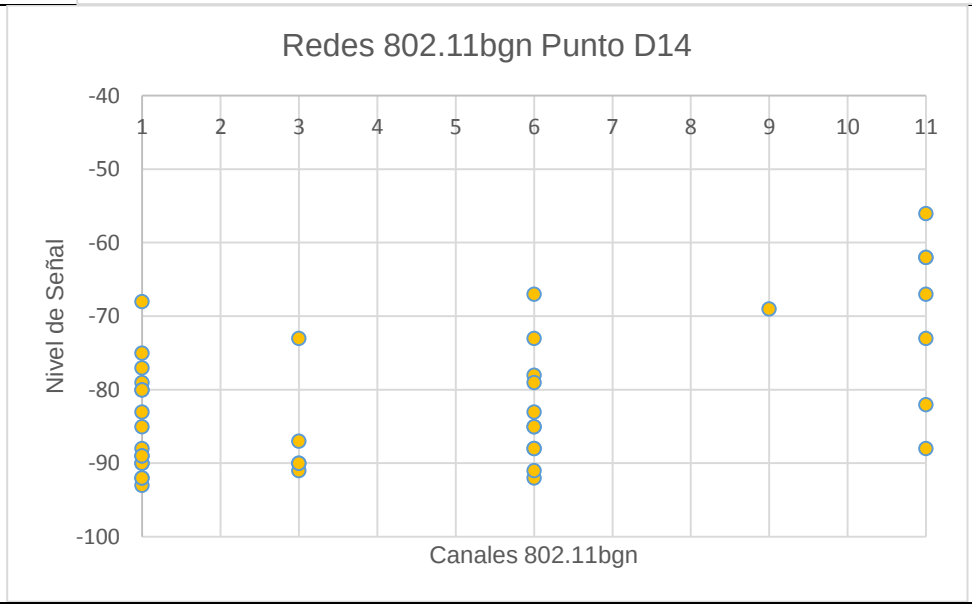
P23		 Redes 802.11bgn Punto D3 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P24		 Redes 802.11bgn Punto D4 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

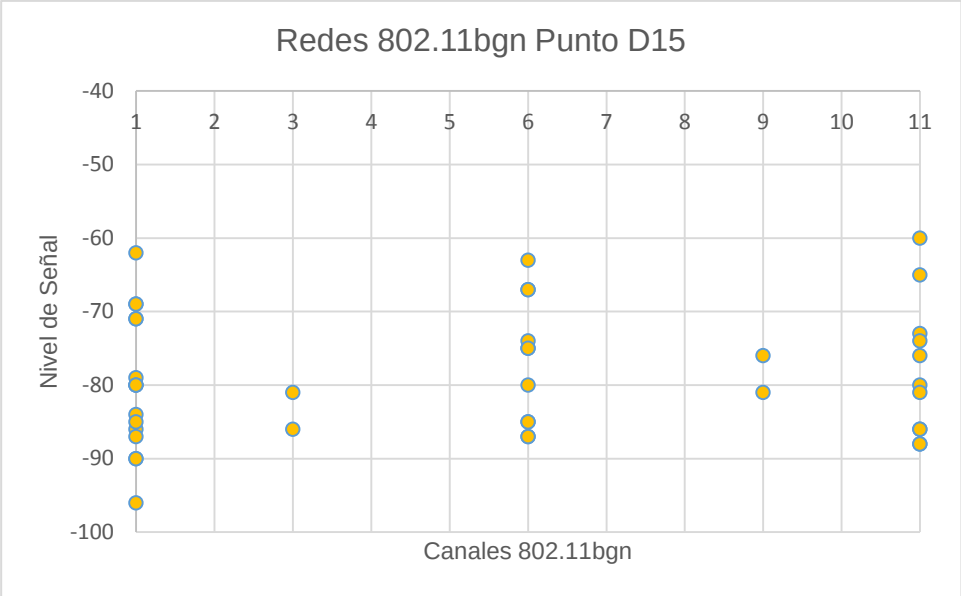
P25	 Redes 802.11bgn Punto D5 Nivel de Señal -50 -55 -60 -65 -70 -75 -80 -85 -90 -95 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P26	 Redes 802.11bgn Punto D6 Nivel de Señal -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

P27	 Redes 802.11bgn Punto D7 Nivel de Señal -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn
P28	 Redes 802.11bgn Punto D8 Nivel de Señal -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Canales 802.11bgn

P29	 Redes 802.11bgn Punto D9 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn
P30	 Redes 802.11bgn Punto D10 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn

P31	 Redes 802.11bgn Punto D11 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn
P32	 Redes 802.11bgn Punto D12 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn

P33	 Redes 802.11bgn Punto D13 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn
P34	 Redes 802.11bgn Punto D14 -40 -50 -60 -70 -80 -90 -100 1 2 3 4 5 6 7 8 9 10 11 Nivel de Señal Canales 802.11bgn

P35	

2. Código del programa en Java utilizado para enviar tramas entre dos radios XBee y calcular tiempos.

```
2.  import javax.comm.*;
3.  import java.io.*;
4.  import javax.comm.*;
5.  import java.io.*;
6.  import java.util.TooManyListenersException;
7.
8.  public class Rs232 implements Runnable, SerialPortEventListener {
9.
10.    private static OutputStream os;
11.    private InputStream is;
12.    private boolean open;
13.    private CommPortIdentifier portId;
14.    private SerialPort sPort;
15.    private SerialParameters parameters;
16.
17.
18.    public Rs232() {
19.       parameters = new SerialParameters("COM11", 9600, 0, 0, 8, 1, 0);
20.       try {
21.          openConnection();
22.       } catch (SerialConnectionException e) {
23.          System.out.println("Error opening port!" + e.getMessage() + ". Select new settings, try again.");
24.          e.printStackTrace();
25.       } catch (Exception e){
26.        e.printStackTrace();
27.       }
28.
29.
30.       Thread thread = new Thread(this);
```

```
31.       thread.start();
32.
33.    }
34.
35.    public void run() {
36.       try {
37.          while (true/*status == DOWNLOADING*/) {
38.             Thread.sleep(5000);
39.          }
40.       } catch (Exception e) {
41.          e.printStackTrace();
42.       } finally {
43.       }
44.    }
45.
46.    public void openConnection() throws SerialConnectionException {
47.
48.       try {
49.          portId = CommPortIdentifier.getPortIdentifier(parameters.getPortName());
50.
51.       } catch (NoSuchPortException e) {
52.          throw new SerialConnectionException(e.getMessage());
53.       } catch (Exception e) {
54.          System.out.println(e.getMessage());
55.       }
56.
57.       try {
58.          sPort = (SerialPort) portId.open("SerialDemo", 30000);
59.          System.out.println("Conexion a puerto Serial OK!");
60.       } catch (PortInUseException e) {
61.          throw new SerialConnectionException(e.getMessage());
62.       }
63.
64.       try {
65.          setConnectionParameters();
66.       } catch (SerialConnectionException e) {
67.          sPort.close();
68.          throw e;
69.       }
70.
71.       try {
72.          os = sPort.getOutputStream();
73.          is = sPort.getInputStream();
74.       } catch (IOException e) {
75.          sPort.close();
76.          throw new SerialConnectionException("Error opening i/o streams");
77.       }
78.
79.       try {
80.          sPort.addEventListener(this);
81.       } catch (TooManyListenersException e) {
82.          sPort.close();
83.          throw new SerialConnectionException("too many listeners added");
84.       }
85.
86.       sPort.notifyOnDataAvailable(true);
87.
88.       sPort.notifyOnBreakInterrupt(true);
89.
90.       try {
91.          sPort.enableReceiveTimeout(30);
```

```
92.         } catch (UnsupportedCommOperationException e) {
93.         }
94.
95.         open = true;
96.     }
97.
98.     public void setConnectionParameters() throws SerialConnectionException {
99.
100.        int oldBaudRate = sPort.getBaudRate();
101.        int oldDatabits = sPort.getDataBits();
102.        int oldStopbits = sPort.getStopBits();
103.        int oldParity = sPort.getParity();
104.        int oldFlowControl = sPort.getFlowControlMode();
105.
106.        try {
107.            sPort.setSerialPortParams(parameters.getBaudRate(),
108.                    parameters.getDatabits(),
109.                    parameters.getStopbits(),
110.                    parameters.getParity());
111.        } catch (UnsupportedCommOperationException e) {
112.            parameters.setBaudRate(oldBaudRate);
113.            parameters.setDatabits(oldDatabits);
114.            parameters.setStopbits(oldStopbits);
115.            parameters.setParity(oldParity);
116.            throw new SerialConnectionException("Unsupported parameter");
117.        }
118.
119.        try {
120.            sPort.setFlowControlMode(parameters.getFlowControlIn()
121.                    | parameters.getFlowControlOut());
122.        } catch (UnsupportedCommOperationException e) {
123.            throw new SerialConnectionException("Unsupported flow control");
124.        }
125.    }
126.
127.    public void closeConnection() {
128.        if (!open) {
129.            return;
130.        }
131.
132.        if (sPort != null) {
133.            try {
134.                os.close();
135.                is.close();
136.            } catch (IOException e) {
137.                System.err.println(e);
138.            }
139.            sPort.close();
140.        }
141.        open = false;
142.    }
143.
144.    public static boolean sendBytes(byte msg[]) {
145.        try {
146.            os.write(msg);
147.            return true;
148.        } catch (IOException e) {
149.            e.printStackTrace();
150.            return false;
151.        }
152.    }
```

```
153.
154.   public boolean isOpen() {
155.      return open;
156.   }
157.
158.   public void serialEvent(SerialPortEvent e) {
159.      StringBuffer inputBuffer = new StringBuffer();
160.      int newData = 0;
161.
162.      switch (e.getEventType()) {
163.
164.         case SerialPortEvent.DATA_AVAILABLE:
165.                  byte[] b ;
166.                  b = new byte[808];
167.                  int i = 0, idMessage=0;
168.
169.            while (newData != -1) {
170.               try {
171.                  newData = is.read();
172.                  if (newData == -1) {
173.                     break;
174.                  }
175.
176.                  if ('\r' != (char) newData) {
177.                     b[i++] = (byte) newData;
178.                  } else {
179.                    byte[] datetime = new byte[8];
180.                    for (int j=0;j<8;j++){
181.                                 datetime[j] = b[j];
182.                    }
183.
184.                    int i1 = b[8]<0 ? 256+b[8] : b[8];
185.                    int i0 = b[9]<0 ? 256+b[9] : b[9];
186.
187.                    idMessage += i1<<8;
188.                    idMessage += i0;
189.
190.                    long now = System.currentTimeMillis();
191.                    long llego = byteArrayToLong(datetime);
192.                     System.out.println(idMessage+","+llego+","+(now - llego));
193.
194.                    i=0;
195.                    b = new byte[808];
196.                    idMessage = 0;
197.                  }
198.
199.               } catch (IOException ex) {
200.                  System.err.println(ex);
201.                  return;
202.               }
203.            }
204.            if (inputBuffer.length() > 0) {
205.               String message = inputBuffer.toString();
206.            }
207.            break;
208.         case SerialPortEvent.BI:
209.      }
210.   }
211.
212.   public long byteArrayToLong(byte[] value) {
213.      long miliseconds=0;
```

```
214.        long b7 = value[7]<0 ? 256+value[7] : value[7];
215.        long b6 = value[6]<0 ? 256+value[6] : value[6];
216.        long b5 = value[5]<0 ? 256+value[5] : value[5];
217.        long b4 = value[4]<0 ? 256+value[4] : value[4];
218.        long b3 = value[3]<0 ? 256+value[3] : value[3];
219.        long b2 = value[2]<0 ? 256+value[2] : value[2];
220.        long b1 = value[1]<0 ? 256+value[1] : value[1];
221.        long b0 = value[0]<0 ? 256+value[0] : value[0];
222.
223.        miliseconds  = b0<<56;
224.        miliseconds += b1<<48;
225.        miliseconds += b2<<40;
226.        miliseconds += b3<<32;
227.        miliseconds += b4<<24;
228.        miliseconds += b5<<16;
229.        miliseconds += b6<<8;
230.        miliseconds += b7;
231.
232.        return miliseconds;
233.    }
234.
235. }
```